

REMARKS

In the Office Action¹, the Examiner provisionally rejected claims 8 and 18 under the judicially created doctrine of obviousness-type doubling patenting as being unpatentable over claims 4, 12, and 19 of copending U.S. Patent Application No. 10/676,374 ("the '374 application"); rejected claims 1-9 and 18-22 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Pub. No. 2002/0108101 to Charisius et al. ("*Charisius*"); and rejected claims 10-17 under 35 U.S.C. § 102(b) as being anticipated by *Charisius*.

Applicant has amended claims 1, 10, and 18, cancelled claims 5 and 13, and added dependent claims 23 and 24. The rejections of cancelled claims 5 and 13 are now moot. Claims 1-4, 6-12, and 14-24 are now pending.

I. PROVISIONAL DOUBLE PATENTING REJECTION

Applicants respectfully traverse the non-statutory double patenting rejection of claims 8 and 18. Applicants request that the Examiner continue to hold the rejection in abeyance for at least the reason that no actual double-patenting circumstance can arise until a patent issues from the present application or the '374 application. Upon review of the remarks made in this paper, should the Examiner believe this application to be in condition for allowance but for the double patenting rejections held in abeyance, Applicants respectfully request that the Examiner contact the undersigned representative to discuss an appropriate resolution.

¹ The Office Action contains a number of statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicant declines to automatically subscribe to any statement or characterization in the Office Action.

II . REJECTION UNDER § 103(a)

Applicants respectfully traverse the rejection of claims 1-9 and 18-22 under 35 U.S.C. § 103(a) as being unpatentable over *Charisius*. A *prima facie* case of obviousness has not been established.

“The key to supporting any rejection under 35 U.S.C. § 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious [R]ejections on obviousness cannot be sustained with mere conclusory statements.” See *M.P.E.P. § 2142, 8th Ed., Rev. 6 (Sept. 2007)*. “The mere fact that references can be combined or modified does not render the resultant combination obvious unless the results would have been predictable to one of ordinary skill in the art” at the time the invention was made. *M.P.E.P. § 2143.01(III), internal citation omitted*. Moreover, “[i]n determining the differences between the prior art and the claims, the question under 35 U.S.C. § 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious.” *M.P.E.P. § 2141.02(I), internal citations omitted (emphasis in original)*.

“[T]he framework for objective analysis for determining obviousness under 35 U.S.C. § 103(a) is stated in *Graham v. John Deere Co.*, 383 U.S. 1, 148 U.S.P.Q 459 (1966) The factual inquiries . . . [include determining the scope and content of the prior art and] . . . [a]scertaining the difference between the claimed invention and the prior art.” *M.P.E.P. § 2141(II)*. “Office personnel must explain why the difference(s) between the prior art and the claimed invention would have been obvious to one of ordinary skill in the art.” *M.P.E.P. § 2141(III)*.

Independent claim 1, as amended, recites, in part,

receive a first model in a first language from a storage device, the first model defining development objects representing building blocks for developing the application, relationships among the development objects, and constraints for developing the application;

generate a set of intermediate objects using the first model, wherein the set of intermediate objects comprises Java objects; and

generate an API using the set of intermediate objects as inputs such that the API enforces the relationships and the constraints defined in the first model and enables accessing the development objects.

The Office Action alleges that *Charisius* discloses, “receiv[ing] a first model in a first language” (Office Action at page 5). The Office Action also alleges that a “graphical view of language representation . . . **reads on** model being in first language” (emphasis in the original) (Office Action at page 5). Even if a “graphical view of language representation” corresponds to a “model being in first language,” it does not demonstrate that *Charisius* teaches or suggests “receiv[ing] a first model in a first language from a storage device,” (emphasis added) as recited in claim 1.

Charisius discloses a “software development tool that creates a graphical representation of source code” (emphasis added) (paragraph 0057). *Charisius* further discloses, the “graphical representation of the project may be in Uniform Modeling Language . . . [a] developer . . . uses the software development tool to open a file which contains [the] . . . source code” (paragraphs 0088-0089). *Charisius* also discloses, the “software development tool generates a transient meta model (TMM) 200 which stores a language neutral representation of the source code 202. The graphical

204 . . . representations of the source code 202 are generated from the language neutral representation in the TMM 200” (paragraphs 0057-0058). Generating a graphical view from language neutral representation in the TMM (model) does not teach or suggest “receiv[ing] a first model in a first language from a storage device,” (emphasis added), as recited in claim 1.

The Office Action also seems to allege that *Charisius* discloses “generat[ing] a set of intermediate objects using the first model,” as recited in claim 1 (Office Action at page 6). This is also not correct.

Charisius discloses, after “opening a file which contains existing source code . . . or creat[ing] a file in which the source code will be developed[,] . . . the software development tool then obtains a template for the current programming language in which the source code is written . . . that can be used to build the data structureThe software development tool uses the template to parse the source code . . . and create the data structure” (paragraph 0089). Fig. 5 of *Charisius* depicts a data structure of the language neutral representation of the source code in Fig. 4 (paragraph 0060). As discussed above, the Office Action alleges that a “graphical view of language representation . . . **reads on** model being in first language” (emphasis in the original) (Office Action at page 5). *Charisius* does not teach or suggest generating based on the “graphical view of language representation.” None of creating a data structure by using a template to parse source code, generating a TMM (model) of the source code, and generating a graphical view from the language neutral representation in the TMM, as disclosed by *Charisius*, teach or suggest “generat[ing] a set of

intermediate (Java) objects using a first model, wherein the set of intermediate objects comprises Java objects” (emphasis added), as recited in claim 1.

The Office Action also seems to allege that *Charisius* discloses, “generat[ing] an API using the set of intermediate objects as inputs,” as recited in claim 1 (Office Action at page 6). This is also not correct.

Charisius discloses, the software development “tool 610 comprises . . . an open application program interface (API), and modules 704There are three main packages composing the API 702: IDE, RWI, and SCI” (emphasis added) (paragraph 0064). For example, “IDE 708 is the API needed to generate custom outputs based on information contained in a model” (paragraph 0065). The Office Action provides other examples of “using the API instance” (Office Action at page 6). The Office Action alleges, “using the packages from derived template and class symbols represented in UML model, along with graphical view of code objects to generate a [] metamodel within interface 610, including instance of RWI, IDE or SCI **reads on** API being instantiated using intermediate objects” (emphasis in the original) (Office Action at page 6).

First, *Charisius* does not disclose or suggest using UML model, along with graphical view of code objects to generate a metamodel within interface 610, including instance of RWI, IDE or SCI, as alleged by the Office Action. *Charisius* discloses that IDE, RWI, and SCI are packages that comprise the API and are not metamodels, as alleged by the Office Action. Furthermore, converting the source code into the language-neutral representation in the TMM (transient meta model) (*Charisius* paragraph 0064), does not constitute or suggest “using . . . UML model, along with

graphical view of code objects to generate a metamodel” (emphasis added), as alleged by the Office Action.

Second, *Charisius* does not disclose or suggest generating an API, as alleged by the Office Action. Rather, the “software development tool comprises” an API composed of an IDE, RWI, and SCI (paragraph 0064). *Charisius* discloses using the API and its components, IDE, RWI, and SCI, as part of the software development tool, **not** generating an API or any of its components. Accordingly, *Charisius* does not teach or suggest “generat[ing] an API using the set of intermediate objects,” as recited in claim 1.

In view of at least the above deficiencies of the *Charisius* reference, the Office Action has neither properly determined the scope and content of the prior art nor properly ascertained the differences between the prior art and the invention of claim 1. Accordingly, the Office Action has failed to clearly articulate a reason why claim 1 would have been obvious to one of ordinary skill in the art in view of the prior art. Therefore, a *prima facie* case of obviousness has not been established with respect to claim 1 and the rejection under 35 U.S.C. § 103(a) must be withdrawn.

Independent claim 18, though of different scope than claim 1, recites similar elements, and is thus allowable over *Charisius* for at least similar reasons as claim 1. Claims 2-4, 9, and 19-22 depend from independent claims 1 and 18, and are thus allowable over *Charisius* for at least the same reasons as the independent claims.

New claims 23 and 24, depend from independent claims 1 and 18, respectively, and are thus also allowable over *Charisius* for at least the same reasons as the independent claims.

III . REJECTIONS UNDER § 102(b)

Applicants respectfully traverse the rejection of claims 10-12 and 14-17 under 35 U.S.C. § 102(b) as being anticipated by *Charisiu*. In order to properly establish that *Charisiu* anticipates Applicants' claimed invention under 35 U.S.C. § 102, each and every element of each of the claims in issue must be found, either expressly described or under principles of inherency, in that single reference. Furthermore, "[t]he identical invention must be shown in as complete detail as is contained in the . . . claim." See M.P.E.P. § 2131, quoting *Richardson v. Suzuki Motor Co.*, 868 F.2d 1126, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989).

Independent claim 10, as amended, recites, for example, "receiv[ing] a first model in a first language from a storage device, the first model defining development objects representing building blocks for developing the application, relationships among the development objects, and constraints for developing the application, wherein the first language comprises unified modeling language." *Charisius* does not disclose at least these elements of claim 1.

Charisius discloses, "the software development tool . . . allow[s] a developer to transform the data structure in the DTD file into an XML structure diagram so the developer can visualize how a class of documents sent from the remote computer are to be defined or interpretedIn addition . . . the software development tool allows the developer to first model the data structure in an XML structure diagram and then generate a corresponding DTD file from the XML structure diagram" (emphasis added). (paragraph 0101). "[R]eceiving documents" written in Extensible Markup Language (XML) (see paragraph 0099 of *Charisius*) does not constitute "receiving a first model in

a first language . . . wherein the first language comprises unified modeling language"
(emphasis added), as recited in claim 10.

Accordingly, for at least the reasons stated above, *Charisius* cannot anticipate claim 10. Claims 11, 12, and 14-17 depend from independent claim 10, and are thus allowable over *Charisius* for at least the same reasons as the independent claim.


In view of the foregoing, Applicants respectfully requests reconsideration of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: October 10, 2008

By: 
Eli Mazour
Reg. No. 59,318
direct telephone: (202) 408 4320